

Group 1
FDM Trainer Skills and Availability

**ECS506U Software Engineering Group
Project**

Requirements Elicitation Report

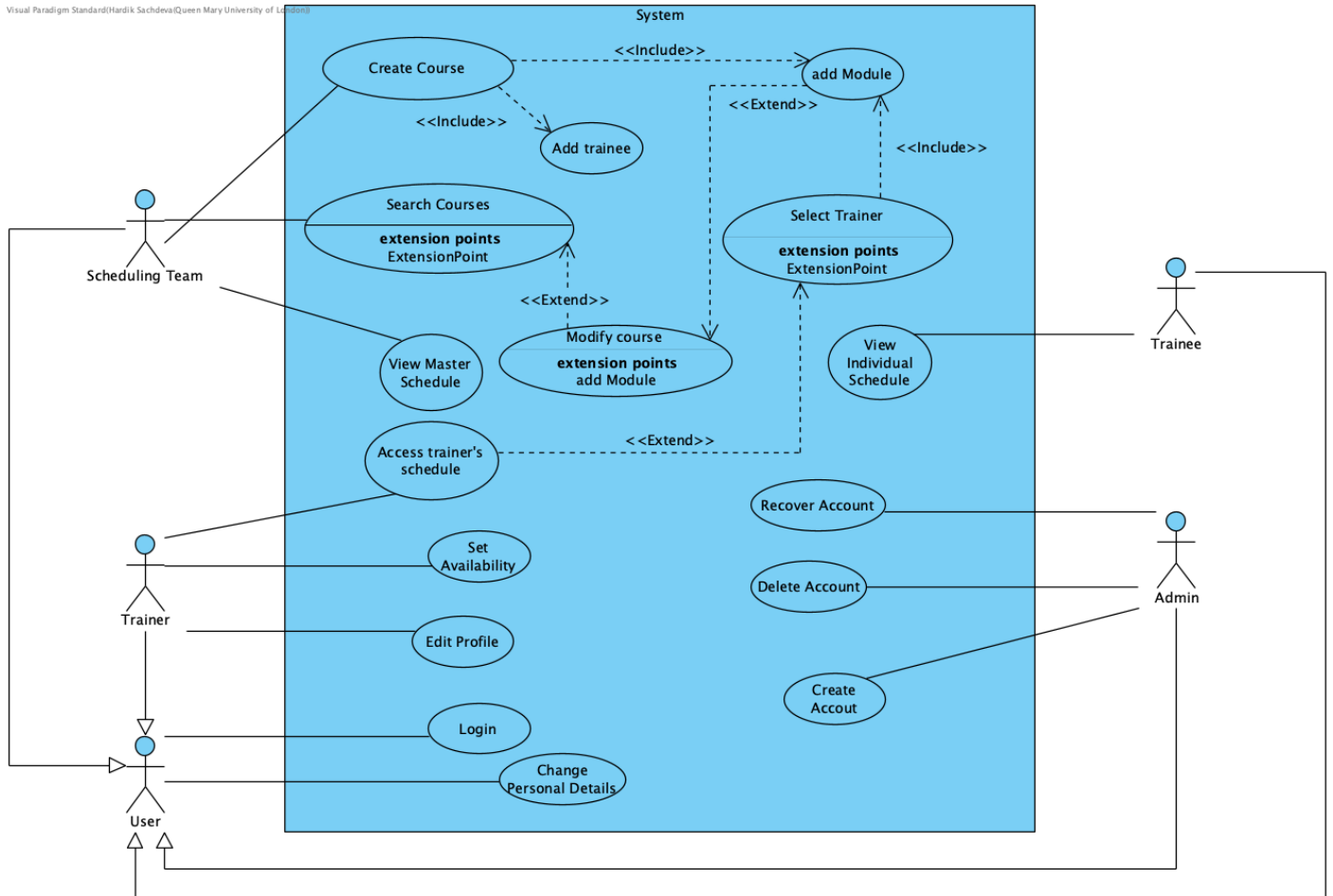
1. List of Requirements

| ID | Requirement | Type | Priority | Use Case |
|----|--|------------------------------|----------|----------|
| 1 | System must meet GDPR compliance rules, especially regarding data privacy | Non-functional Security | Core | - |
| 2 | Schedule page's response time should not exceed 3 seconds | Non-functional Performance | Optional | - |
| 3 | All timestamps recorded into the system's database shall be in Greenwich Mean Time Zone (GMT) | Non-functional Usability | Optional | - |
| 4 | System must be capable of running on the latest versions of Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge | Non-functional Accessibility | Core | - |
| 5 | System must not allow performance of any use cases without authentication | Non-functional Security | Core | - |
| 6 | Password must be 8 characters long containing at least one upper case letter, one lower case letter and a special character. | Non-Functional Security | Optional | - |
| 7 | System's database should be capable of storing up to 10,000 users | Non-Functional Capacity | Core | - |
| 8 | System must implement initial two factor authentication via email/SMS prior to first login | Non-functional Security | Optional | - |
| 9 | Allow the users to setup security questions so that they can reset their password | Non-functional Security | Optional | - |
| 10 | System must be able to process 500 concurrent users | Non-functional Capacity | Core | - |
| 11 | All text must have high contrast with background in order to be easily readable | Non-Functional Accessibility | Optional | - |
| 12 | Only scheduling members and trainers must be able to view a trainer schedule. | Non-Functional Usability | Core | - |
| 13 | System must classify room as booked for the duration of the module for which it is reserved. | Non-Functional | Core | - |
| 14 | Web application should be adaptable to mobile browsers | Non-Functional Usability | Core | - |
| 15 | System must not allow to create a course without any modules. | Non-Functional Usability | Core | - |
| 16 | System must not allow to create a module without a trainer. | Non-Functional Usability | Core | - |
| 17 | System must not allow booking rooms for a module that has no trainees. | Non-Functional Usability | Core | - |
| 18 | A trainee cannot view a trainer schedule. | Non-Functional Usability | Core | - |
| 19 | System must not allow the creation of a course without any modules | Non-Functional | Core | - |
| 20 | System must ensure all users have a unique username | Functional (D) | Core | Login |

| | | | | |
|----|--|----------------|----------|---|
| 21 | All users must be able to log in to the system with a password. | Functional (D) | Core | Login |
| 22 | System must allow users to change password | Functional | Optional | Login |
| 23 | System should allow scheduling team members to be able to create courses. | Functional | Core | Create Course |
| 24 | Scheduling team members must add modules to courses | Functional (D) | Core | Create course |
| 25 | Scheduling team members must be able to search trainers with specific skills. | Functional | Optional | Create course |
| 26 | System must allow scheduling team members to search for room availability. | Functional | Core | Create course |
| 27 | System must ensure scheduling team member adds a room to the modules where they will be taught. | Functional (D) | Core | Create course |
| 28 | System should ensure each course has a unique ID and a description. | Functional (D) | Core | Create course |
| 29 | System must ensure each room has a unique ID and must store the location of each room. | Functional (D) | Core | Create course |
| 30 | When a trainer or trainee have been added to a module or a course they must be notified by email. | Functional | Optional | Create course |
| 31 | Course scheduling team members must be able to choose to book a course for an online meeting instead of a room | Functional | Core | Create course |
| 32 | Scheduling team member must be able to create modules. | Functional | Core | Add module |
| 33 | System should ensure each module has a unique ID and a description as well. | Functional (D) | Core | Add module |
| 34 | Scheduling team members must be able to search for trainer's time schedules and view them. | Functional | Core | Access trainer's schedule |
| 35 | Scheduling team members must be able to search for trainer's available at a certain time. | Functional | Optional | Select Trainer |
| 36 | System must allow scheduling team members to add trainers to each module. The trainer must teach this module. | Functional (D) | Core | Select Trainer |
| 37 | When a trainer is added to a module, the system must show that module on the trainer's schedule. | Functional | Core | Select Trainer |
| 38 | System must ensure scheduling team members add trainees to a course. | Functional (D) | Core | Add Trainee |
| 39 | Scheduling team members must be able to view a trainee's schedule and search for their availability. | Functional | Optional | Add Trainee |
| 40 | System must show all modules for a course a trainee has been added to, on that trainee's schedule. | Functional | Core | Add Trainee |
| 41 | Each trainee and trainer must have a separate and unique schedule. | Functional (D) | Core | Access trainer's schedule/ View individual schedule |
| 42 | System must include a master schedule which includes scheduling for all courses and all modules. | Functional | Core | View Master Schedule |
| 43 | Only scheduling team can view the master schedule. | Functional | Core | View Master Schedule |
| 44 | System must only allow the scheduling team members to edit the master schedule and the trainee schedule | Functional | Core | View Master Schedule/Access trainer's schedule |
| 45 | Trainers must be able to add available hours to their own trainer schedule. | Functional (D) | Core | Set Availability |

| | | | | |
|----|--|----------------|----------|--|
| 46 | Trainers must be able to add skills they can teach to their profiles. | Functional (D) | Optional | Edit Profile |
| 47 | System must allow trainers to view and print their schedule. | Functional (D) | Core | Access trainer's Schedule |
| 48 | System must allow trainees to view and print their schedule. | Functional (D) | Core | View Individual Schedule |
| 49 | If a module is taking place online, then the schedule for that module must show a link to the meeting room. This must be shown for both trainer and trainee schedules. | Functional (D) | Optional | Access trainer's schedule/View Individual Schedule |
| 50 | A trainee must add an email and a phone number to their account. | Functional (D) | Optional | Change Personal Details |
| 51 | A trainer must add an email and a phone number to their account. | Functional (D) | Optional | Change Personal Details |
| 52 | Users must be able to change their personal details. | Functional | Optional | Change Personal Details |
| 53 | Scheduling team members can reschedule or change the location of a module. | Functional | Optional | Modify course |
| 54 | When a module information is changed all trainee's and trainers involved with the module must be notified with an email. | Functional | Optional | Modify course |
| 55 | System must allow trainees can view which trainer teaches which module. | Functional (D) | Core | View Individual Schedule |
| 56 | Trainees must be able to check the rooms a module is booked in. | Functional (D) | Core | View Individual Schedule |
| 57 | System must allow trainers can check the rooms a module is booked in. | Functional (D) | Core | Access trainer's schedule |
| 58 | System must allow trainers to declare the reason for non-availability in their timetable. | Functional (D) | Optional | Set Availability |
| 59 | Scheduling team members can search for existing modules and add them to a course. | Functional (D) | Optional | Add Module |
| 60 | Admin must be able to recover account | Functional | Optional | Recover Account |
| 61 | Admin must be able to create new user accounts | Functional | Core | Create Account |
| 62 | Admin must be able to delete user accounts | Functional | Core | Delete Account |
| 63 | System must store first name and last name for all trainers and trainees. | Functional (D) | Core | Create Account |

2. Use-Case Diagram



3. Use Case Description

In this section, describe one of the key use cases from the diagram in the previous section. Only include the following information.

Name – Add Module.

Brief Description – Adds module to a course. Multiple modules can be added to a course.

Actors – Scheduling team members

Preconditions – A course must already exist (have been created by scheduling team members) for modules to be added to it, user must be signed in as scheduling team member and on course page

Basic Flow

1. The module needs to be created, so scheduling team member clicks on the create new module button.
2. System opens window for creating a new module.
3. The user inputs the name of the module, it's unique ID, designated timeslots during desired business day (i.e., Wednesdays 9-11 am) and writes a brief description about the module.
4. System checks to confirm whether module with same ID or Name already exists
5. The user requests to search trainers with specific skills by typing required skills in a search bar.
6. The system displays all trainers who are available at time of module and possess the required skills
7. The user selects a trainer and allocates the module to the predetermined timeslot in the trainer's schedule.
8. System checks there aren't any conflicts with the trainees' schedules that are enrolled in the course.
9. System takes user to next tab to book a room for the module. System displays all rooms available at the time of the module.
10. User selects room to be booked for the module.
11. System confirms the module has been added to the course and takes user back to the course page.

Alternate Flows

After step 4:

- **5.1** System cannot confirm that module with same Name or ID already exists in database
- **5.2** System alerts the user that module already exists in database and returns them to step

After step 5:

- **6.1** There is at least 1 available and qualified trainer, they are displayed
- **6.2** Trainers with required skill/skills do not exist or are not available at scheduled module time slots. System notifies user "No qualified and available trainer available" and returns user to step 5

After step 8:

- **9.1** System does not detect a conflict and takes them to the book room tab
- **9.2** System detects a conflict in a trainee's schedule with the new module, so it notifies the user and gives them the option to continue or return them to step 3 to select a new time

After step 9:

- **10.1** User selects room, and the confirmation is made
- **10.2** No rooms are available at the time of the module, returns user to step 3 to select new time

Post Conditions

- The trainer and trainees are sent an email notifying them about the module they have been assigned to or enrolled in
- The module shows up on the trainers' and all the trainees' individual schedules. The schedule displays the trainer, designated room for module, date and time.
- The chosen room is classified as "not available" at the time of the module
- The module is also added to "master schedule" which contains all courses and modules

4. Risk Assessment

| Risk | Likelihood | Severity | Impact | Preventative / Mitigating Actions |
|--|------------|----------|--|--|
| Poor time management | High | High | Unfinished product. | Evaluate progress frequently and adjust project plan if necessary |
| Missing member(s) due to health issues | High | Medium | Unfinished/ low quality product. | Have frequent meetings and communication to make sure all members aware of each other's progress. |
| Project team lack of understanding of the requirements | Medium | High | Product doesn't support requirements. | Have very accurate documentation for each step of the development. |
| Client contact leaves the company | Low | Medium | Product is delayed. | Establish a second representative from our client's company which could replace the first. |
| Devices used by the development team aren't stable | Low | Medium | Product is delayed/ progression is lost. | Equip the development team with reliable devices and make sure to backup data regularly. |
| Client has erroneous expectations from the project | Low | High | Client is unsatisfied. | Have frequent meetings with the client so that they are on par with the development and can give feedback. |
| System is unable to be used in the organisation | Low | High | Unusable final product. | Communicate clearly with the stakeholders the intent and function of the application. |
| Technologies used in development are not compatible | Low | High | Incomplete product. | Research all technologies that will be used during development and ensure they are compatible and adjust if they are not. |
| Development environment problems | Medium | High | Product finish date is delayed. | Confirm that each member of the development team is comfortable with the environment and establish a second development environment as a backup. |
| Incorrect requirement priorities | High | Medium | Product is delayed. | Create a clear list of requirements with their respective priorities so that the development team is always focusing on the most important requirements first. |